(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2003－16410
（P2003－16410A）
(43)公開日　平成15年１月17日(2003.1.17)

(51)Int.Cl.[7]		識別記号	FI		テーマコード(参考)	
Ｇ０６Ｋ	19/07		Ｂ４２Ｄ	15/10	５２１	２Ｃ００５
Ｂ４２Ｄ	15/10	５２１	Ｇ０６Ｋ	17/00	Ｄ	５Ｂ０３５
Ｇ０６Ｆ	9/44			19/00	Ｎ	５Ｂ０５８
	9/54		Ｇ０６Ｆ	9/06	６４０Ｂ	５Ｂ０７６
Ｇ０６Ｋ	17/00				６２０Ｋ	

審査請求　未請求　請求項の数13　ＯＬ　（全 13 頁）

(21)出願番号　特願2001－201807(P2001－201807)

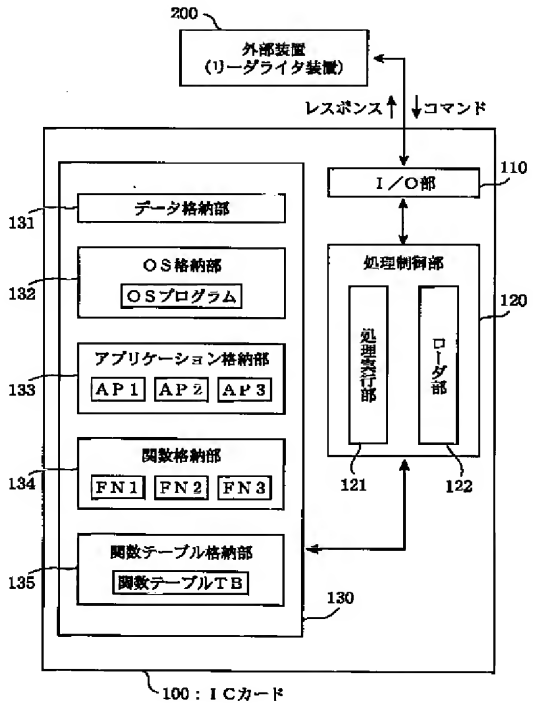
(22)出願日　平成13年７月３日(2001.7.3)

(71)出願人　000002897
大日本印刷株式会社
東京都新宿区市谷加賀町一丁目１番１号
(72)発明者　柴田　直人
東京都新宿区市谷加賀町一丁目１番１号
大日本印刷株式会社内
(72)発明者　入澤　和義
東京都新宿区市谷加賀町一丁目１番１号
大日本印刷株式会社内
(74)代理人　100091476
弁理士　志村　浩

最終頁に続く

(54)【発明の名称】　ＩＣカード

(57)【要約】
【課題】　複数のプログラムから共通して利用可能な関数モジュールを、単体として容易に追加する。
【解決手段】　ローダ部１２２に、アプリケーションおよび関数モジュールをロードする機能を設ける。新たにロードされたアプリケーションはアプリケーション格納部１３３に格納され、新たにロードされた関数モジュールは関数格納部１３４に格納される。ロードされた関数モジュールの関数名と実アドレスとの対応関係を示す関数テーブルＴＢが関数テーブル格納部１３５に作成される。アプリケーションからの関数呼び出しは、関数テーブルＴＢを参照して、実アドレスを認識することにより行われる。関数モジュールは、ＯＳプログラムや他の関数モジュールからも呼び出される。



【特許請求の範囲】
【請求項１】　外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行するＩＣカードであって、
外部装置との交信を行うＩ／Ｏ部と、
データを格納するデータ格納部と、
ＯＳプログラムを格納するＯＳ格納部と、
アプリケーションプログラムを格納するアプリケーション格納部と、
関数モジュールを格納する関数格納部と、
前記関数格納部に格納されている関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係を示す関数テーブルを格納する関数テーブル格納部と、
外部装置からロードコマンドとともにアプリケーションプログラムが与えられた場合に、与えられたアプリケーションプログラムを前記アプリケーション格納部に格納する処理と、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、与えられた関数モジュールを前記関数格納部に格納するとともに当該格納した関数モジュールについての関数テーブルを前記関数テーブル格納部に格納する処理と、を前記ＯＳプログラムに基づいて実行するローダ部と、
外部装置から与えられたコマンドを、前記ＯＳプログラムまたは前記アプリケーションプログラムに基づいて実行し、実行結果をレスポンスとして外部装置へと返す処理を行う処理実行部と、
を備え、
前記処理実行部は、
外部装置から与えられたアプリケーション選択コマンドに基づいて、もしくはＯＳ格納部に格納されているＯＳプログラムで指示された初期設定に基づいて、前記アプリケーション格納部に格納されている複数のアプリケーションプログラムのうちの１つを選択する機能と、
外部装置からアプリケーション用コマンドが与えられた場合に、当該コマンドを現時点で選択されているアプリケーションプログラムに基づいて実行する機能と、
実行中のアプリケーションプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、前記関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、前記実行中のアプリケーションプログラムを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数モジュールを実行する機能と、
を有することを特徴とするＩＣカード。
【請求項２】　請求項１に記載のＩＣカードにおいて、処理実行部が、実行中のＯＳプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、前記関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、前記実行中のＯＳプログラムを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数モジュールを実行する機能を有することを特徴とするＩＣカード。
【請求項３】　請求項１に記載のＩＣカードにおいて、処理実行部が、実行中の関数モジュールから、特定の関数名を指定した関数実行命令が与えられた場合に、前記関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、前記実行中の関数モジュールを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数を実行する機能を有することを特徴とするＩＣカード。
【請求項４】　請求項１に記載のＩＣカードにおいて、外部装置から「カプセル化された関数実行命令を含むＯＳ用もしくはアプリケーション用コマンド」が与えられた場合に、処理実行部が、当該コマンドをＯＳプログラムもしくは現時点で選択されているアプリケーションプログラムに基づいて実行し、関数テーブル格納部内の関数テーブルを参照することにより、前記関数実行命令によって示されている特定の関数名に対応する実アドレスを認識し、前記ＯＳプログラムもしくは現時点で選択されているアプリケーションプログラムを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数モジュールを実行する機能を有することを特徴とするＩＣカード。
【請求項５】　請求項１～４のいずれかに記載のＩＣカードにおいて、
処理実行部が、特定の関数モジュールを実行する際に、関数の呼出元となるプログラムにおけるプログラムカウンタを退避する処理を行い、前記特定の関数モジュールの実行後に、退避したプログラムカウンタに応じた位置から、前記呼出元となるプログラムの実行に復帰することを特徴とするＩＣカード。
【請求項６】　請求項１～５のいずれかに記載のＩＣカードにおいて、
外部装置からロードコマンドおよび関数モジュールとともに、当該関数モジュールの呼出元を限定する呼出元限定情報が与えられた場合に、ローダ部が、当該関数モジュールについての前記呼出元限定情報を関数テーブルの一部として格納する処理を行う機能を有し、
処理実行部が、前記関数テーブル内の呼出元限定情報によって示された限定条件の下で関数モジュールの実行を行うことを特徴とするＩＣカード。
【請求項７】　請求項１～５のいずれかに記載のＩＣカードにおいて、
ローダ部が、現時点でＩＣカード内に格納されている呼出元となるプログラムから呼び出しが可能な関数名を外部装置にレスポンスとして報知する機能を有することを特徴とするＩＣカード。
【請求項８】　請求項１～５のいずれかに記載のＩＣカ

ードにおいて、
外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、与えられた関数モジュールを呼び出すための呼出元となるプログラムが現時点でＩＣカード内に格納されているか否かを調査し、格納されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返す機能を有することを特徴とするＩＣカード。
【請求項９】　請求項１～５のいずれかに記載のＩＣカードにおいて、
外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、所定のセキュリティ条件が満足されているか否かを調査し、満足されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返す機能を有することを特徴とするＩＣカード。
【請求項１０】　請求項１～５のいずれかに記載のＩＣカードにおいて、
外部装置からロードコマンドとともに暗号化された関数モジュールが与えられた場合に、ロード部が、前記暗号化された関数モジュールを復号化してから関数格納部に格納する機能を有することを特徴とするＩＣカード。
【請求項１１】　請求項１～５のいずれかに記載のＩＣカードにおいて、
外部装置からロードコマンドとともに署名付きの関数モジュールが与えられた場合に、ロード部が、前記署名が正しいことを検証した後に、前記関数モジュールを関数格納部に格納する機能を有することを特徴とするＩＣカード。
【請求項１２】　請求項１～５のいずれかに記載のＩＣカードにおいて、
ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールの利用を一時停止させる機能と、この一時停止からの復帰をさせる機能と、を有することを特徴とするＩＣカード。
【請求項１３】　請求項１～５のいずれかに記載のＩＣカードにおいて、
ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールを削除、置換、または変更する機能を有することを特徴とするＩＣカード。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明はＩＣカードに関し、特に、外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行するＩＣカードに関する。
【０００２】
【従来の技術】携帯可能な情報記録媒体として、ＩＣカードが様々な分野で利用されるようになってきている。現在、一般に普及しているＩＣカードは、ＣＰＵを内蔵し、外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行する機能を有している。どのようなコマンドを与えると、ＩＣカード内部でどのような処理が実行されるかは、ＩＣカードに組み込まれたＯＳプログラムおよびアプリケーションプログラムによって定まるため、通常、個々の用途に応じた処理機能を有する専用のアプリケーションプログラムがＩＣカード内に組み込まれる。最近では、半導体回路の集積度がかなり高まり、ＩＣカードに内蔵されるメモリ容量も飛躍的に向上してきたため、複数のアプリケーションプログラムを予め組み込んだ多用途向けのＩＣカードも一般化してきている。また、発行後に、必要に応じて新たなアプリケーションプログラムを追加する機能を備えたＩＣカードも広く利用されている。たとえば、特開平１０－７９０００号公報には、バージョンアップなどのためにプログラムを追加する機能をもったＩＣカードが開示されている。
【０００３】
【発明が解決しようとする課題】上述したように、新たなアプリケーションプログラムを、必要に応じて、その都度、追加することができるＩＣカードは、既に利用されている。しかしながら、このようなアプリケーションプログラムは、あくまでも単体で機能するプログラムであり、従来のＩＣカードにおいて追加できるプログラムは、このような単体で機能するプログラムに限られている。ところが、実際には、関数モジュールのように、単体では機能せずに、ＯＳプログラムやアプリケーションプログラムなどから１つのルーチンとして利用されるプログラムも少なくない。従来、このような関数モジュールは、ＯＳプログラムやアプリケーションプログラムの中に組み込まれており、関数モジュールを単体としてＩＣカードに追加することは困難である。もちろん、アプリケーションプログラム自体に、このような関数モジュールを追加する何らかの機能をもたせておけば、当該アプリケーションからは、追加した関数モジュールを利用することはできる。しかしながら、個々のアプリケーションごとにこのような機能を用意することは非効率的であり、あまり実用的な方法とは言えない。
【０００４】そこで本発明は、複数のプログラムから共通して利用可能な関数モジュールを、単体として容易に追加することが可能なＩＣカードを提供することを目的とする。
【０００５】
【課題を解決するための手段】(1)　本発明の第１の態様は、外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行するＩＣカード

において、外部装置との交信を行うＩ／Ｏ部と、データを格納するデータ格納部と、ＯＳプログラムを格納するＯＳ格納部と、アプリケーションプログラムを格納するアプリケーション格納部と、関数モジュールを格納する関数格納部と、関数格納部に格納されている関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係を示す関数テーブルを格納する関数テーブル格納部と、外部装置からロードコマンドとともにアプリケーションプログラムが与えられた場合に、与えられたアプリケーションプログラムをアプリケーション格納部に格納する処理と、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、与えられた関数モジュールを関数格納部に格納するとともに当該格納した関数モジュールについての関数テーブルを関数テーブル格納部に格納する処理と、をＯＳプログラムに基づいて実行するローダ部と、外部装置から与えられたコマンドを、ＯＳプログラムまたはアプリケーションプログラムに基づいて実行し、実行結果をレスポンスとして外部装置へと返す処理を行う処理実行部と、を設け、処理実行部が、外部装置から与えられたアプリケーション選択コマンドに基づいて、もしくはＯＳ格納部に格納されているＯＳプログラムで指示された初期設定に基づいて、アプリケーション格納部に格納されている複数のアプリケーションプログラムのうちの１つを選択する機能と、外部装置からアプリケーション用コマンドが与えられた場合に、当該コマンドを現時点で選択されているアプリケーションプログラムに基づいて実行する機能と、実行中のアプリケーションプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のアプリケーションプログラムを呼出元として、関数格納部内の実アドレスに格納されている関数モジュールを実行する機能と、を行うことができるようにしたものである。

【０００６】(2)　本発明の第２の態様は、上述の第１の態様に係るＩＣカードにおいて、処理実行部が、実行中のＯＳプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のＯＳプログラムを呼出元として、関数格納部内の実アドレスに格納されている関数モジュールを実行するようにしたものである。

【０００７】(3)　本発明の第３の態様は、上述の第１の態様に係るＩＣカードにおいて、処理実行部が、実行中の関数モジュールから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中の関数モジュールを呼出元として、関数格納部内の実アドレスに格納されている関数を実行するようにしたものである。

【０００８】(4)　本発明の第４の態様は、上述の第１の態様に係るＩＣカードにおいて、外部装置から「カプセル化された関数実行命令を含むＯＳ用もしくはアプリケーション用コマンド」が与えられた場合に、処理実行部が、当該コマンドをＯＳプログラムもしくは現時点で選択されているアプリケーションプログラムに基づいて実行し、関数テーブル格納部内の関数テーブルを参照することにより、関数実行命令によって示されている特定の関数名に対応する実アドレスを認識し、ＯＳプログラムもしくは現時点で選択されているアプリケーションプログラムを呼出元として、関数格納部内の実アドレスに格納されている関数モジュールを実行するようにしたものである。

【０００９】(5)　本発明の第５の態様は、上述の第１～第４の態様に係るＩＣカードにおいて、処理実行部が、特定の関数モジュールを実行する際に、関数の呼出元となるプログラムにおけるプログラムカウンタを退避する処理を行い、特定の関数モジュールの実行後に、退避したプログラムカウンタに応じた位置から、呼出元となるプログラムの実行に復帰できるようにしたものである。

【００１０】(6)　本発明の第６の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、外部装置からロードコマンドおよび関数モジュールとともに、当該関数モジュールの呼出元を限定する呼出元限定情報が与えられた場合に、ローダ部が、当該関数モジュールについての呼出元限定情報を関数テーブルの一部として格納する処理を行い、処理実行部が、関数テーブル内の呼出元限定情報によって示された限定条件の下で関数モジュールの実行を行うようにしたものである。

【００１１】(7)　本発明の第７の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、ローダ部が、現時点でＩＣカード内に格納されている呼出元となるプログラムから呼び出しが可能な関数名を外部装置にレスポンスとして報知するようにしたものである。

【００１２】(8)　本発明の第８の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、与えられた関数モジュールを呼び出すための呼出元となるプログラムが現時点でＩＣカード内に格納されているか否かを調査し、格納されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返すようにしたものである。

【００１３】(9)　本発明の第９の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、所定のセキュリティ条件が満足され

ているか否かを調査し、満足されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返すようにしたものである。

【００１４】(10)　本発明の第１０の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、外部装置からロードコマンドとともに暗号化された関数モジュールが与えられた場合に、ロード部が、暗号化された関数モジュールを復号化してから関数格納部に格納するようにしたものである。

【００１５】(11)　本発明の第１１の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、外部装置からロードコマンドとともに署名付きの関数モジュールが与えられた場合に、ロード部が、署名が正しいことを検証した後に、関数モジュールを関数格納部に格納するようにしたものである。

【００１６】(12)　本発明の第１２の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールの利用を一時停止させる機能と、この一時停止からの復帰をさせる機能と、を有するようにしたものである。

【００１７】(13)　本発明の第１３の態様は、上述の第１～第５の態様に係るＩＣカードにおいて、ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールを削除、置換、または変更する機能を有するようにしたものである。

【００１８】

【発明の実施の形態】以下、本発明を図示する実施形態に基づいて説明する。

【００１９】§１．本発明に係るＩＣカードの基本構成

図１は、本発明の一実施形態に係るＩＣカード１００を、外部装置２００に接続した状態を示すブロック図である。外部装置２００は、一般に、リーダライタ装置と呼ばれている装置であり、通常は、パソコンなどに接続して利用される。ＩＣカード１００には、外部装置２００と交信を行うためのＩ／Ｏ部１１０と、ＩＣカード内で種々の処理や制御を行う処理制御部１２０と、種々のデータやプログラムを格納するメモリ部１３０とが内蔵されている。ＩＣカード１００と外部装置２００との間の交信は、コマンドおよびレスポンスのやりとりによって行われる。すなわち、外部装置２００側から所定のコマンドをＩＣカード１００に対して送信すると、このコマンドは、Ｉ／Ｏ部１１０において受信されて処理制御部１２０へと伝達される。処理制御部１２０は、このコマンドを解釈実行し、処理結果をレスポンスとしてＩ／Ｏ部１１０へと引き渡す。こうして、このレスポンスは、Ｉ／Ｏ部１１０から外部装置２００へと送信されることになる。処理制御部１２０は、メモリ部１３０内に格納されているプログラムに基づいて、与えられたコマンドの実行を行い、このコマンドの実行に伴い、必要に応じて、メモリ部１３０内に格納されているデータへのアクセスを行うことになる。

【００２０】ここに示す実施形態では、処理制御部１２０は、処理実行部１２１とローダ部１２２とによって構成されている。また、メモリ部１３０は、データ格納部１３１、ＯＳ格納部１３２、アプリケーション格納部１３３、関数格納部１３４、関数テーブル格納部１３５の５つの格納部によって構成されている。もっとも、これらの各構成要素は、本発明を説明する便宜上、機能面に着目して捉えた構成要素であり、実際のハードウエア上の構成要素に対応しているものではない。実際には、処理制御部１２０は、ＩＣカード内に埋め込まれたＣＰＵやその他の論理装置によって実現され、メモリ部１３０は、ＲＯＭ，ＲＡＭ，ＥＥＰＲＯＭなどのメモリによって実現されることになる。また、処理制御部１２０を構成する処理実行部１２１およびローダ部１２２は、いずれもＣＰＵ単独で実現できる構成要素ではなく、メモリ部１３０内に格納されているプログラムに基づく動作が前提となって実現される構成要素である。ただ、ここでは、本発明の構成を説明する上で、ＣＰＵによって実行される機能のうち、外部から与えられた一般的なコマンドを実行する構成要素を処理実行部１２１とし、外部から与えられた新たなアプリケーションプログラムや関数モジュールをメモリ部１３０内に格納するロードコマンドを実行する構成要素をローダ部１２２として把握することにする。

【００２１】メモリ部１３０の構成要素のうち、データ格納部１３１は、このＩＣカード１００内に格納すべき種々のデータを格納する部分であり、このＩＣカード１００の所有者であるユーザに関する個人データや取引データ、ＩＣカード１００の発行者に関するデータなどが格納される。これらのデータは、通常、不揮発性の書き込み可能なメモリであるＥＥＰＲＯＭに格納される。また、ここでは、このデータ格納部１３１内に、いわゆるＣＰＵの作業領域も設けられており、種々の変数、ポインタ、フラグなども、このデータ格納部１３１内に格納されるものとする。このような作業領域は、通常、揮発性の書き込み可能なメモリであるＲＡＭ内に設けられる。

【００２２】一方、ＯＳ格納部１３２には、このＩＣカード１００の基本動作を記述したＯＳプログラムが格納される。現在、ＩＣカード用のＯＳプログラムとしては、ＪａｖａＣａｒｄやＭＵＬＴＯＳなどが普及しており、処理制御部１２０は、このＯＳプログラムに基づいて基本的な動作を行うことになる。このようなＯＳ格納部１３２は、ＲＯＭ内のメモリ領域あるいはＥＥＰＲＯＭ内のメモリ領域に設けられる。これに対して、アプリ

ケーション格納部１３３には、アプリケーションプログラムが格納される。ここに示す実施形態の場合、アプリケーションプログラムはＩＣカードの発行後に任意に追加することができる仕様となっており、アプリケーション格納部１３３はＥＥＰＲＯＭ内のメモリ領域に設けられている。図１に示す例では、３つのアプリケーションプログラムＡＰ１，ＡＰ２，ＡＰ３が既にアプリケーション格納部１３３に格納された状態が示されている。

【００２３】関数格納部１３４および関数テーブル格納部１３５は、本発明の特徴となるメモリ領域であり、ＥＥＰＲＯＭ内のメモリ領域に設けられている。関数格納部１３４には、単独のプログラムとしては機能しない関数モジュールが格納される。図１に示す例では、３つの関数モジュールＦＮ１，ＦＮ２，ＦＮ３が既に関数格納部１３４に格納された状態が示されている。関数テーブル格納部１３５には、この関数格納部１３４内に格納されている関数モジュールを利用するために必要となる関数テーブルＴＢが格納される。この関数テーブルＴＢには、関数格納部１３４に格納されている関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係が示されている。

【００２４】処理実行部１２１は、外部装置２００から与えられた一般のコマンド（後述するように、「ローダ部１２２によって実行されるべきロードコマンド」以外のコマンド）を、ＯＳ格納部１３２内のＯＳプログラムまたはアプリケーション格納部１３３内のアプリケーションプログラムに基づいて実行し、実行結果をレスポンスとして外部装置２００へと返す処理を行うことになる。外部装置２００側から与えるコマンドは、ＯＳプログラムによって解釈実行可能なＯＳ用コマンドと、特定のアプリケーションプログラムによって解釈実行可能なアプリケーション用コマンドと、に分類することができる。ＯＳ用コマンドの典型的な例は、アプリケーション選択コマンドである。アプリケーション選択コマンドは、アプリケーション格納部１３３内に格納されている複数のアプリケーションプログラムのうちの１つを選択するためのコマンドであり、外部装置２００側から、特定のアプリケーションを指定したアプリケーション選択コマンドが与えられると、処理実行部１２１は、指定されたアプリケーションプログラムを選択状態にする。具体的には、当該アプリケーションプログラムが現在選択状態にあることを示す情報を、データ格納部１３１内に格納する。一方、外部装置２００側から、アプリケーション用コマンドが与えられた場合、処理実行部１２１は、このコマンドをＯＳプログラムではなく、アプリケーションプログラムに基づいて処理することになる。このとき、処理実行部１２１は、いずれのアプリケーションプログラムが選択状態にあるかを判断した上で、現在選択状態にあるアプリケーションプログラムに基づいて、与えられたアプリケーション用コマンドを処理する。なお、ＩＣカード１００と外部装置２００とを接続した初期状態においては、ＯＳプログラムによって予め指定された特定のアプリケーションプログラムが、デフォルトの選択プログラムとなる初期設定がなされており、アプリケーション選択コマンドが与えられていない初期状態では、デフォルト指定された特定のアプリケーションプログラムが自動的に選択状態になっている。

【００２５】結局、外部装置２００側からコマンドが与えられた場合、当該コマンドがＯＳ用コマンドであった場合には、ＯＳプログラムに基づいて処理が実行され、当該コマンドがアプリケーション用コマンドであった場合には、現在選択状態にあるアプリケーションプログラムに基づいて処理が実行されることになる。いずれの場合にも、処理の結果を示すレスポンスが作成され、外部装置２００側へ返される。実際には、処理制御部１２０は、ＪａｖａＣａｒｄやＭＵＬＴＯＳなどのＯＳプログラムのもとで動作するアプリケーションプログラムを解釈実行する仮想機械としての機能を果たすことになる。

【００２６】このように、外部装置２００側から与えられた一般のコマンドは、処理実行部１２１によって処理されるが、ロードコマンドについては、ローダ部１２２による処理が行われる。このロードコマンドは、ＯＳプログラムによって実行されるべきＯＳ用コマンドの一種であり、新たなアプリケーションプログラムや新たな関数モジュールをメモリ部１３０内に格納するためのコマンドである。たとえば、図１に示す例では、アプリケーション格納部１３３内には、既に３つのアプリケーションプログラムＡＰ１，ＡＰ２，ＡＰ３が格納された状態となっているが、ここに、更に新たなアプリケーションプログラムＡＰ４を追加する必要が生じた場合には、外部装置２００から所定のロードコマンドとともにアプリケーションプログラムＡＰ４を構成するデータを、ＩＣカード１００側に与える操作を行えばよい。このようなロードコマンドが与えられると、ローダ部１２２によって、このアプリケーションプログラムＡＰ４をアプリケーション格納部１３３内に新たに格納する処理が行われ、これを実行するために必要な設定が行われる。このように、ＩＣカードに、新たにアプリケーションプログラムを追加する処理は、従来の一般的なＩＣカードにおいても既に実施されている公知の技術であり、ここでは、詳しい説明は省略する。

【００２７】本発明の特徴は、ローダ部１２２に、アプリケーションプログラムの追加処理機能だけではなく、単体としては機能しない関数モジュールの追加処理機能を設けた点にある。たとえば、図１に示す例では、関数格納部１３４内には、既に３つの関数モジュールＦＮ１，ＦＮ２，ＦＮ３が格納された状態となっているが、ここに、更に新たな関数モジュールＦＮ４を追加する必要が生じた場合には、外部装置２００から所定のロードコマンドとともに関数モジュールＦＮ４を構成するデー

タをＩＣカード１００側に与える操作を行えばよい。このようなロードコマンドが与えられると、ローダ部１２２によって、この関数モジュールＦＮ４を関数格納部１３４内に新たに格納する処理が行われ、更に、この新たに格納した関数モジュールＦＮ４についての情報を関数テーブル格納部１３５内の関数テーブルＴＢにつけ加える処理が行われることになる。もちろん、このようなアプリケーションプログラムの追加格納処理や、関数モジュールの追加格納処理は、ＯＳプログラムに基づいて実行されるべき処理であり、ローダ部１２２にこのような処理を実行させるためには、ＯＳ格納部１３２内に用意されたＯＳプログラムに、そのような処理を実行させるための手順を記述しておく必要がある。

【００２８】§２．本発明に係るＩＣカードにおける関数の実行

さて、これまで本発明に係るＩＣカードの基本構成について述べた。ここでは、このＩＣカードにおける関数の実行形態について説明を行う。一般に、ＪａｖａＣａｒｄやＭＵＬＴＯＳなど、一般的なＩＣカード用ＯＳプログラムのもとで動作するアプリケーションプログラムは、ＣＰＵに依存しない言語で記述されているため、アプリケーションプログラム自体では、プログラムやファイルを選択するコマンドを取り扱うことが困難である。本発明の狙いは、このように、ＣＰＵに依存しない言語で記述されているアプリケーションプログラムを実行するＩＣカードにおいて、単体としては機能しない関数モジュールを容易に追加することができるようにする点にある。そのためには、関数モジュールを追加する処理および関数モジュールを呼び出す処理を行う機能を、ＯＳプログラムによって提供する必要がある。関数モジュールを追加する処理が、ＯＳプログラムの下で、ローダ部１２２によって実行されることは、既に§１で説明したとおりである。ここでは、関数モジュールを呼び出す処理（関数を実行するための処理）が、ＯＳプログラムの下で、処理実行部１２１によってどのように実行されるかについて説明する。

【００２９】まず、従来の一般的なＩＣカードにおける関数モジュールの取り扱いと、本発明に係るＩＣカードにおける関数モジュールの取り扱いとの概念的な相違を、図２および図３を参照して説明しよう。図２は、従来のＩＣカードにおける関数モジュールの取り扱いを示すブロック図である。図示の例では、２つのアプリケーションプログラムＡＰ１，ＡＰ２がブロックとして示されている。また、各アプリケーションプログラムの中には、それぞれ関数モジュールＦＮ１，ＦＮ２が組み込まれている。このように、従来のＩＣカードの場合、アプリケーションプログラムが関数を利用する場合、当該関数モジュールは、アプリケーションプログラムの中に記述され、アプリケーションプログラムと一体化された形式になる。したがって、ローダ部１２２によって、アプリケーションプログラムをアプリケーション格納部１３３内に格納した場合、当該アプリケーションプログラムで利用する関数モジュールも、このアプリケーションプログラムと一体となってアプリケーション格納部１３３内に格納されることになる。このように、従来のＩＣカードでは、関数モジュールがアプリケーションプログラムの一部として融合した形になっていたため、ＯＳプログラムには、関数モジュールを単独で取り扱う仕組みが用意されていない。

【００３０】これに対して、本発明の基本思想では、図３に示す例のように、個々の関数モジュールをアプリケーションプログラム本体とは別個のプログラムとして、単独で取り扱う仕組みを用意しておき、必要に応じて必要な関数モジュールを、その都度、呼び出して利用する、という手法を採る。図３の例の場合、２つのアプリケーションプログラムＡＰ１，ＡＰ２と、３つの関数モジュールＦＮ１，ＦＮ２，ＦＮ３とが用意されており、アプリケーションプログラムＡＰ１から関数モジュールＦＮ１が呼び出され、アプリケーションプログラムＡＰ２から関数モジュールＦＮ２が呼び出されている様子が示されている。関数モジュールＦＮ１の本体は、アプリケーションプログラムＡＰ１内には含まれておらず、別個独立したプログラムとして関数格納部１３４内に格納されており、関数モジュールＦＮ２の本体は、アプリケーションプログラムＡＰ２内には含まれておらず、こちらも別個独立したプログラムとして関数格納部１３４内に格納されている。

【００３１】このように、本発明の場合、アプリケーションプログラムも関数モジュールも、ローダ部１２２によって独立してロードされるプログラムであるという点では同じである。ただ、アプリケーションプログラムは、単体で何らかの機能を果たすプログラムであるのに対し、関数モジュールは、単体では機能せず、他のプログラムから呼び出されることによってその役目を果たすプログラムである、という点で相違しており、両者のこの相違点は、外部装置２００側からの選択コマンドによる選択対象になるか否かという点における相違に結びつくことになる。すなわち、アプリケーションプログラムは、前述したように、外部装置２００側からアプリケーション選択コマンドを与えることにより選択することができ、外部から明示的に特定のアプリケーションの実行を指示することができるが、関数モジュールは、このような外部からの選択コマンドにより選択して実行させることはできず、あくまでも他のプログラムから呼び出して利用することが前提となる。

【００３２】アプリケーションプログラムから、特定の関数モジュールを呼び出すには、呼出元となるアプリケーションプログラム内に、特定の関数の実行を指示する関数実行命令を記述しておくようにすればよい。たとえば、図４に示すように、アプリケーションプログラムＡ

Ｐ１と、関数モジュールＦＮ１，ＦＮ２とがロードされている場合を考えてみよう。ここでは、アプリケーションプログラムＡＰ１は、アプリケーション格納部１３３内のアドレスＡｄｄ１～Ａｄｄ５の領域に格納されており、関数モジュールＦＮ１は、関数格納部１３４内のアドレスＡｄｄ６～Ａｄｄ７の領域に格納されており、関数モジュールＦＮ２は、関数格納部１３４内のアドレスＡｄｄ８～Ａｄｄ９の領域に格納されているものとする。図示の例では、アプリケーションプログラムＡＰ１には、３か所に関数実行命令（この例では、「ＣＡＬＬ　ＦＮ１」のような文字列）が含まれている。これらの関数実行命令は、特定の関数名を指定して、これを実行する旨の命令であり、たとえば、アドレスＡｄｄ２に記述された「ＣＡＬＬ　ＦＮ１」なる命令は、「ＦＮ１」なる関数名を指定した関数実行命令である。

【００３３】既に§１で述べたように、ローダ部１２２が新たな関数モジュールを関数格納部１３４内に格納する処理を行う際には、当該関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係を示す関数テーブルＴＢを、関数テーブル格納部１３５に格納する処理も行われる。図４の右下に示す関数テーブルＴＢは、関数モジュールＦＮ１およびＦＮ２について作成された関数テーブルＴＢの具体例を示すものである。この例では、関数名ＦＮ１に対応する実アドレスはＡｄｄ６となっており、関数モジュールＦＮ１が格納されている先頭実アドレスがＡｄｄ６であることが示されており、また、関数名ＦＮ２に対応する実アドレスはＡｄｄ８となっており、関数モジュールＦＮ２が格納されている先頭実アドレスがＡｄｄ８であることが示されている。処理実行部１２１は、実行中のアプリケーションプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部１３５内の関数テーブルＴＢを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のアプリケーションプログラムを呼出元として、関数格納部１３４内の認識した実アドレスに格納されている関数モジュールを実行することになる。

【００３４】たとえば、アプリケーションプログラムＡＰ１内のルーチンを実行中に、アドレスＡｄｄ２に記述された「ＣＡＬＬ　ＦＮ１」なる関数実行命令に遭遇した場合には、関数テーブルＴＢを参照することにより、関数名ＦＮ１に対応する実アドレスがアドレスＡｄｄ６であることを認識し、このアドレスＡｄｄ６に格納されている関数モジュールＦＮ１のルーチンの実行（呼び出し）を行うことになる。一般に、関数モジュールは、呼出元から与えられた何らかの変数Ｘ（複数の場合もあるし、文字列の場合もある）に基づいて、関数値Ｆ（Ｘ）を演算してこれを呼出元へと返す機能を果たすプログラムである。したがって、実際には、このアプリケーションプログラムＡＰ１のアドレスＡｄｄ２に至るまでのルーチンにおいて、変数Ｘとして何らかの値が用意され、アドレスＡｄｄ２以降のルーチンにおいて、関数モジュールＦＮ１から返された関数値Ｆ（Ｘ）を利用した何らかの処理が行われることになる。ここで、アプリケーションプログラムから関数モジュールへの変数Ｘの引き渡しや、関数モジュールからアプリケーションプログラムへの関数値Ｆ（Ｘ）の引き渡しは、たとえば、レジスタなどを用いた公知の方法で行うことができるので、ここでは詳しい説明は省略する。

【００３５】アプリケーションプログラムから関数モジュールを呼び出した後は、再びアプリケーションプログラムへと復帰することになるが、このような復帰は、プログラムカウンタの退避処理によって行うことができる。すなわち、処理実行部１２１は、関数モジュールＦＮ１を実行する際に、この関数の呼出元となるアプリケーションプログラムＡＰ１におけるプログラムカウンタ（アドレスＡｄｄ２を示すカウンタ）を退避する処理を行う。そして、呼出先となった関数モジュールＦＮ１の実行が完了した後に、退避したプログラムカウンタに応じた位置（アドレスＡｄｄ２にある関数実行命令の次の命令の位置）から、呼出元となったアプリケーションプログラムＡＰ１の実行に復帰することになる。

【００３６】なお、図４に示すアプリケーションプログラムＡＰ１のアドレスＡｄｄ４には、関数モジュールＦＮ３についての関数実行命令が記述されているが、この図４に示す例では、関数モジュールＦＮ３はまだ関数格納部１３４内には格納されていない。したがって、図示の例においてアドレスＡｄｄ４に記述された関数実行命令は、現時点では実行不能な命令であり、プログラムカウンタがアドレスＡｄｄ４を指し示した場合には、エラーが生じてしまう。実は、この図４に示すアプリケーションプログラムＡＰ１におけるアドレスＡｄｄ４以降のルーチンは、将来の機能拡張用に予め付加されているルーチンであり、実際には、この時点では、このルーチンが実行されることはない。別言すれば、このルーチンが実行されるようなコマンドは、この時点では、ＩＣカードに対して与えない、という前提で、外部装置２００側からのアクセスが行われることになる。もちろん、将来、アプリケーションプログラムＡＰ１の機能を拡張する必要が生じた場合には、ロードコマンドを用いて関数モジュールＦＮ３を新たにＩＣカード内に格納すれば、このアドレスＡｄｄ４以降のルーチンによる付加機能が利用できるようになる。

【００３７】このように、本発明に係るＩＣカードでは、必要に応じて、その都度、必要な関数モジュールを新規にロードして利用できるため、柔軟な利用形態が可能になる。また、同一の関数モジュールを複数のアプリケーションプログラムから共通して利用するような形態も可能になり、関数の効率的な利用を図ることもできるようになる。

【００３８】§３．本発明における関数モジュールの利用形態の変形例
以上、本発明に係るＩＣカードを基本的な実施形態に基づいて説明したが、本発明は上述した基本的な実施形態に限定されるものではなく、この他にも種々の形態で実施可能である。ここでは、本発明における関数モジュールの利用形態の変形例のいくつかを項分けして紹介する。
【００３９】(1)　上述の実施形態では、アプリケーションプログラムから関数モジュールを呼び出す例のみを示したが、関数モジュールの呼出元は、必ずしもアプリケーションプログラムである必要はなく、たとえば、ＯＳプログラムを呼出元として特定の関数モジュールを呼び出すようなことも可能である。すなわち、処理実行部１２１は、実行中のＯＳプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部１３５内の関数テーブルＴＢを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のＯＳプログラムを呼出元として、関数格納部１３４内の所定の実アドレスに格納されている関数モジュールを実行する処理を行えばよい。また、第１の関数モジュールを呼出元として更に第２の関数モジュールを呼び出すような、いわゆる入れ子式の関数呼び出しを行うことも可能である。すなわち、処理実行部１２１は、実行中の関数モジュールから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部１３５内の関数テーブルＴＢを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中の関数モジュールを呼出元として、関数格納部１３４内の所定の実アドレスに格納されている関数を実行する処理を行えばよい。
【００４０】(2)　このように、本発明では、関数格納部１３４内の関数モジュールは、ＩＣカード内部のプログラム（アプリケーションプログラム、ＯＳプログラム、他の関数モジュール）から呼び出されることが前提となっており、原則として、外部装置２００側から関数格納部１３４内の関数モジュールを直接呼び出して利用することはできない。ただ、外部装置２００側から与えるコマンドに工夫を施せば、あたかも外部装置２００側から関数格納部１３４内の関数モジュールを間接的に呼び出して利用するかのような利用形態も可能である。具体的には、外部装置２００側から、「カプセル化された関数実行命令を含むＯＳ用もしくはアプリケーション用コマンド」をＩＣカード１００側に与えるようにすればよい。そして、処理実行部１２１は、このようなコマンドが与えられたときに、当該コマンドを、ＯＳプログラムもしくは現時点で選択されているアプリケーションプログラムに基づいて実行し、関数テーブル格納部１３５内の関数テーブルＴＢを参照することにより、カプセル化されていた関数実行命令によって示されている特定の関数名に対応する実アドレスを認識し、ＯＳプログラムもしくは現時点で選択されているアプリケーションプログラムを呼出元として、関数格納部１３４内の所定の実アドレスに格納されている関数モジュールを実行する処理を行うようにすればよい。
【００４１】図５は、このような処理を実行させるためのアプリケーション用コマンドの構成例を示すブロック図である。このコマンドは、命令コードを示す「ＥＸＥＣＦＮ」なる文字列の部分と、当該命令コードに付随するデータとなる「ＣＡＬＬＦＮ１，Ｘ＝１２３」なる文字列の部分と、によって構成されている。ここで、後半の「ＣＡＬＬ　ＦＮ１，Ｘ＝１２３」なる部分は、実際には、関数実行命令および変数値を示すコードであるが、カプセル化された状態となっており、全体として見れば、「ＥＸＥＣＦＮ」なる命令コードのデータ部分という形式になっている。したがって、このようなアプリケーション用コマンドが外部装置２００側から与えられると、一般のコマンドと同様に、ＯＳプログラムもしくはその時点で選択状態となっているアプリケーションプログラムへと引き渡されることになる。そこで、ＯＳプログラムもしくはアプリケーションプログラム側に、この「ＥＸＥＣＦＮ」なるコマンドが与えられた場合に、そのデータ部に記述された「ＣＡＬＬ　ＦＮ１，Ｘ＝１２３」なる文字列を関数実行命令および変数値を示すコードと解釈して、関数モジュールＦＮ１の呼び出し処理を実行するルーチンを予め用意しておけば、関数モジュールＦＮ１に変数値Ｘ＝１２３を引き渡し、関数値Ｆ（１２３）を得ることができる。更に、このアプリケーションプログラム側に、得られた関数値Ｆ（１２３）をレスポンスとして返すルーチンを用意しておけば、図５に示すようなアプリケーション用コマンドに対するレスポンスとして、関数値Ｆ（１２３）が外部装置２００側に返されることになる。
【００４２】このような手法を採れば、実際には、ＩＣカード内に格納されているプログラムから関数モジュールＦＮ１を呼び出して利用していることになるが、実質的には、外部装置２００側からのコマンドによって、関数モジュールＦＮ１を利用するような取り扱いが可能になる。
【００４３】(3)　図４に示す関数テーブルＴＢには、各関数モジュールについて、関数名と実アドレスとの対応関係しか示されていないが、更に、当該関数モジュールの呼出元を限定する呼出元限定情報を付加することも可能である。図６は、このような呼出元限定情報を付加した関数テーブルＴＢの一例を示す図である。たとえば、関数モジュールＦＮ１については、呼出元限定情報として、ＡＰ１，ＡＰ２，ＯＳなる情報が付加されているが、これは、関数モジュールＦＮ１の呼出元を、アプリケーションプログラムＡＰ１，ＡＰ２およびＯＳプログラムに限定することを示す情報であり、関数モジュー

ルＦＮ２については、呼出元限定情報として、ＡＰ１，ＡＰ３なる情報が付加されているが、これは、関数モジュールＦＮ２の呼出元を、アプリケーションプログラムＡＰ１，ＡＰ３に限定することを示す情報である。このような呼出元限定情報は、個々の関数モジュールをロードする際に、外部装置２００側から指定すればよい。ローダ部１２２は、外部装置２００から関数モジュールとともに、当該関数モジュールの呼出元を限定する呼出元限定情報が与えられた場合に、当該関数モジュールについての呼出元限定情報を関数テーブルの一部として格納する処理を行うことになる。

【００４４】処理実行部１２１は、関数テーブルＴＢに呼出元限定情報が付加されていた場合には、この呼出元限定情報によって示された限定条件の下で、関数モジュールの実行を行うことになる。たとえば、図６に示す例の場合、アプリケーションプログラムＡＰ２が関数モジュールＦＮ２を呼び出すような処理を行ったとしても、呼出元限定情報による限定条件を満たしていないので、このような呼出処理は正常な処理とは認められず、エラー処理がなされることになる。このような呼出元限定情報を設定しておくと、特定のアプリケーションプログラムと密接に結び付いた関数モジュールが、他のアプリケーションプログラムから利用されるのを防ぐことができる。

【００４５】§４．本発明における関数モジュールのロード時の変形例

本発明に係るＩＣカードでは、ローダ部１２２によって、アプリケーションプログラムのみならず、必要に応じて関数モジュールを新たにロードすることができることは既に述べたとおりである。ここでは、この関数モジュールのロードを行う際の変形例のいくつかを項分けして紹介する。

【００４６】(1)　既に述べたように、本発明においてＩＣカード内にロードされる関数モジュールは、ＩＣカード内のプログラム（アプリケーションプログラム、ＯＳプログラム、あるいは他の関数モジュール）から呼び出されて利用されることが前提となる。したがって、呼出元となるプログラムがＩＣカードに存在しないような関数モジュールは、ＩＣカード内にロードしても利用価値はない。そこで、ローダ部１２２に、現時点でＩＣカード内に格納されている呼出元となるプログラムから呼び出しが可能な関数名を外部装置にレスポンスとして報知する機能をもたせておくのが好ましい。たとえば、図４に示すようなアプリケーションプログラムＡＰ１のみがＩＣカード内に格納されている状態では、このアプリケーションプログラムＡＰ１内を調べることにより、３つの関数モジュールＦＮ１，ＦＮ２，ＦＮ３のみが呼び出し可能であることが認識できる。そこで、ローダ部１２２に、この３つの関数モジュールの関数名である「ＦＮ１，ＦＮ２，ＦＮ３」なる文字列を、外部装置２００側に報知する機能をもたせておけば、外部装置２００側は、現時点でこのＩＣカードにおいて利用価値のある関数モジュールは、この３つの関数モジュールである旨を認識することができ、誤って利用価値のない関数モジュールをロードするような作業が行われるのを防ぐことができる。ローダ部１２２から外部装置２００側への報知は、レスポンスという形式で行われるが、このレスポンスを得るためには、外部装置２００側から、当該レスポンスを要求する旨のコマンドを与えるようにすればよい。あるいは、ＩＣカード１００と外部装置２００とを接続した初期状態において、ＩＣカードをリセットしたときにＩＣカード側から返されるＡＴＲ（Answer to Reset ）の信号に、呼び出しが可能な関数名を示す情報をもたせておくようにしてもよい。

【００４７】(2)　上述した工夫と類似した工夫として、外部装置２００からロードコマンドとともに関数モジュールが与えられた場合に、ロード部１２２が、与えられた関数モジュールを呼び出すための呼出元となるプログラムが現時点でＩＣカード内に格納されているか否かを調査し、格納されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置２００に返すような処理を行うようにしてもよい。この場合は、ＩＣカード１００側から積極的に、呼び出しが可能な関数名の報知が行われるわけではないが、もし誤って呼び出し不能な関数モジュールをロードするような作業が行われた場合には、エラーが発生することになる。たとえば、図４に示すようなアプリケーションプログラムＡＰ１のみがＩＣカード内に格納されている状態において、新たに関数モジュールＦＮ４をロードするような作業が行われた場合（すなわち、外部装置２００側から、関数モジュールＦＮ４についてのロードコマンドが与えられた場合）は、アプリケーションプログラムＡＰ１内を調べることにより、関数モジュールＦＮ４の呼び出しが不能であることが認識できるので、このようなロードコマンドは実行されずに、エラーを示すレスポンスが返されることになる。

【００４８】(3)　これまでの実施形態では、新たに関数モジュールをロードする際のセキュリティ確保の問題については触れていなかったが、実用上は、外部装置２００からロードコマンドとともに関数モジュールが与えられた場合に、ロード部１２２が、所定のセキュリティ条件が満足されているか否かを調査し、満足されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置２００に返すようにしておくのが好ましい。セキュリティ条件が満足されているか否かの判断は、既に公知の種々の方法のいずれを用いてもかまわない。

【００４９】また、関数モジュールをロードする際には、関数モジュール本体に対する暗号化を行っておくようにするのが好ましい。この場合、外部装置２００側の

暗号化アルゴリズムに応じた復号化アルゴリズムをＩＣカード１００側に用意しておき、外部装置２００からロードコマンドとともに暗号化された関数モジュールが与えられた場合に、ロード部１２２が、用意されている復号化アルゴリズムにより、暗号化された関数モジュールを復号化してから関数格納部１３４に格納する処理を行うようにすればよい。

【００５０】更に、関数モジュールをロードする際には、関数モジュール本体に署名を付加するようにし、ＩＣカード内に、この署名を検証するアルゴリズムを用意しておくのが好ましい。この場合、外部装置２００からロードコマンドとともに署名付きの関数モジュールが与えられた場合に、ロード部１２２が、検証用アルゴリズムを用いて、付加されてきた署名が正しいことを検証した後に、この関数モジュールを関数格納部１３４に格納する処理を行えばよい。

【００５１】(4)　ローダ部１２２には、更に付加的な機能を設けておくこともできる。たとえば、外部装置２００からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部１３４内に格納されている特定の関数モジュールの利用を一時停止させる機能と、この一時停止からの復帰をさせる機能と、をローダ部１２２に設けておくこともできる。この機能を利用すれば、一時的に特定の関数モジュールの利用を制限するような処理を容易に行うことができる。また、外部装置２００からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部１３４内に格納されている関数モジュールを削除、置換、または変更する機能を設けておくこともできる。不要になった関数モジュールが発生した場合、これを削除すれば、その分のメモリ領域を開放することができる。また、新たな関数モジュールに置換したり、部分的に変更したりすることができれば、関数モジュールを最新のバージョンに保つことができるようになる。

【００５２】

【発明の効果】以上のとおり、本発明に係るＩＣカードによれば、複数のプログラムから共通して利用可能な関数モジュールを、単体として容易に追加することが可能になる。

【図面の簡単な説明】

【図１】本発明の一実施形態に係るＩＣカード１００を、外部装置２００に接続した状態を示すブロック図である。

【図２】従来のＩＣカードにおける関数モジュールの取り扱いを示すブロック図である。

【図３】本発明に係るＩＣカードにおける関数モジュールの取り扱いを示すブロック図である。

【図４】アプリケーションプログラムＡＰ１と、関数モジュールＦＮ１，ＦＮ２とがロードされている場合のＩＣカードのメモリ領域の構成を示すブロック図である。

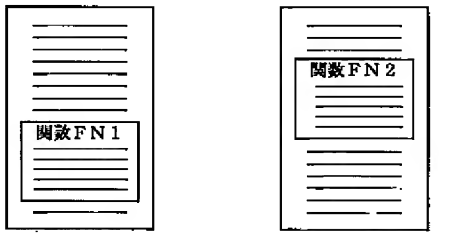
【図５】カプセル化された関数実行命令を含むアプリケーション用コマンドの構成例を示す図である。

【図６】呼出元限定情報を付加した関数テーブルＴＢの一例を示す図である。

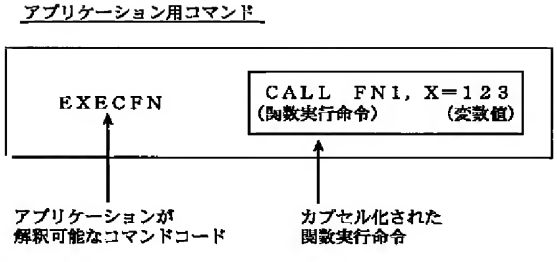
【符号の説明】

１００…ＩＣカード
１１０…Ｉ／Ｏ部
１２０…処理制御部
１２１…処理実行部
１２２…ローダ部
１３０…メモリ部
１３１…データ格納部
１３２…ＯＳ格納部
１３３…アプリケーション格納部
１３４…関数格納部
１３５…関数テーブル格納部
Ａｄｄ１～Ａｄｄ９…実アドレス
ＡＰ１～ＡＰ３…アプリケーションプログラム
ＦＮ１～ＦＮ３…関数モジュール
ＴＢ…関数テーブル

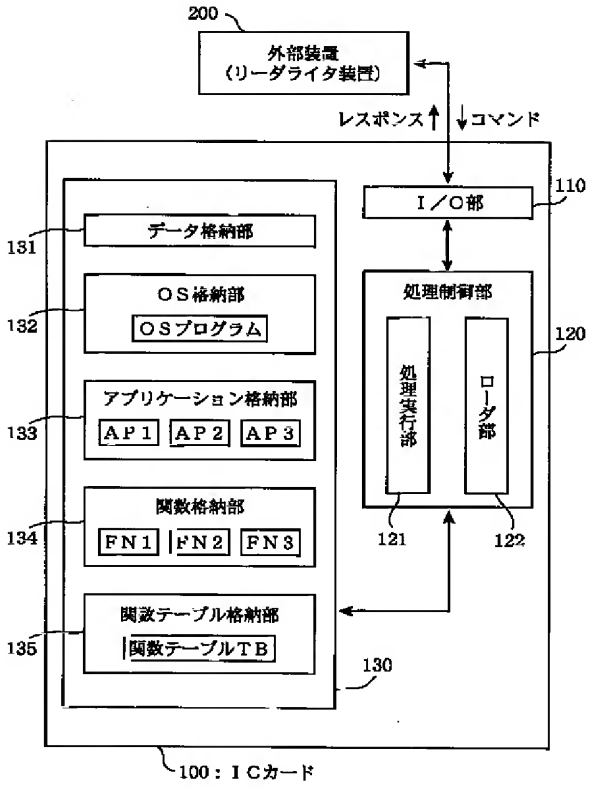
【図２】



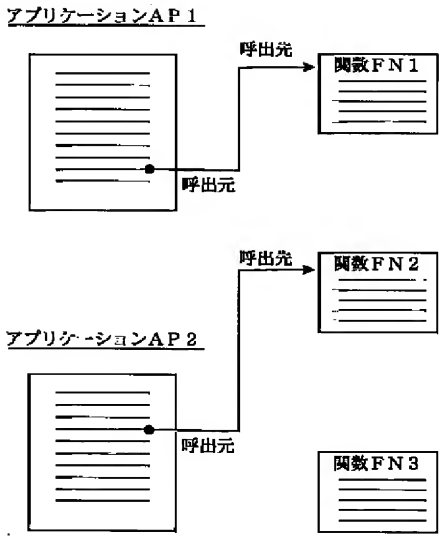
【図５】



【図1】



【図3】

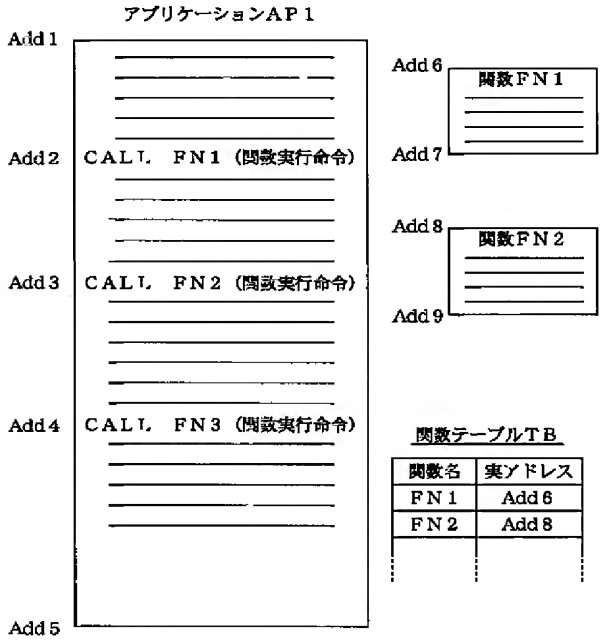


【図6】

関数テーブルTB

関数名	実アドレス	呼出元限定情報
FN1	Add6	AP1，AP2，OS
FN2	Add8	AP1，AP3

【図4】



関数テーブルTB

関数名	実アドレス
FN1	Add6
FN2	Add8

フロントページの続き

(72)発明者　神力　哲夫
東京都新宿区市谷加賀町一丁目１番１号
大日本印刷株式会社内

Ｆターム(参考)　2C005 MA33 SA02 SA04 SA08 SA23
5B035 AA06 BB09 CA11
5B058 CA25 KA02 KA04 KA08 KA11
YA20
5B076 AB02 AB04 AB05 AB06